Apigee Token Sharing Workflow

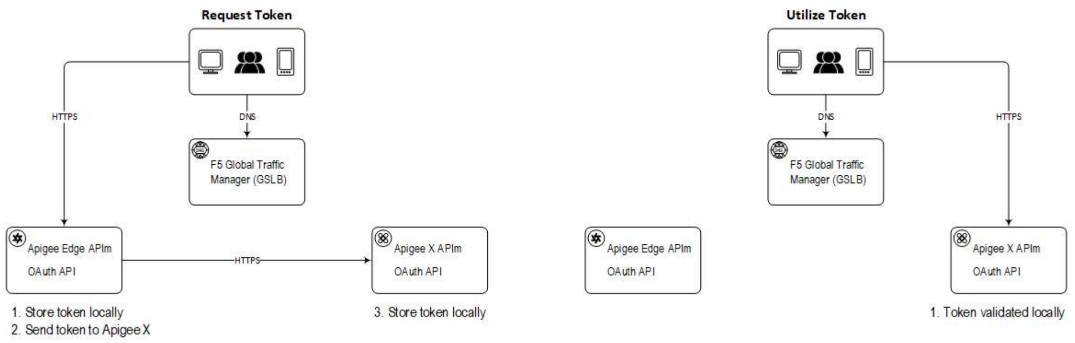
Created by St Louis, Jeff (US - Ohio), last modified by Gullapalli, Ramya (Remote - CAN Ontario) on Feb 15, 2023

As part of the migration we need to manage the tokens which are generated in Apigee Edge and Apigee X. During the migration, we will have to route traffic to both the Apigee Edge and Apigee X environments in a distributed manner. As a result, we will need to share the tokens that are generated in Apigee Edge with the Apigee X environment, and vice versa.

For the token sharing, we are handling the following scenarios:

Tokens Generated in Apigee Edge and Shared with Apigee X

In this scenario, the token generated in Apigee Edge is available in both Apigee Edge and Apigee X to validate API calls for both environments.



When the token request is routed to the OAuth proxy in Apigee Edge, the proxy in Apigee Edge will generate the token, return the token to the client, and then send the token to Apigee X to store the token. Apigee X will use the GenerateOAuth policy external authorization feature to store the token generated by Apigee Edge into Apigee X for the given client ID. When Apigee X receives a request, it now has the token stored to authorize the requested transactions.

As part of the proof of concept, we implemented this approach and we have collected the following metrics for this transaction processing.

The metrics when calling the store token process as part of the generate flow are:

- Total time taken to generate the token: 213ms (120ms, 319ms, 107ms)
- Time taken in Apigee Edge for the service callout to store the token in Apigee X: 190ms
- Time taken in Apigee X for the Store Token service: 109ms

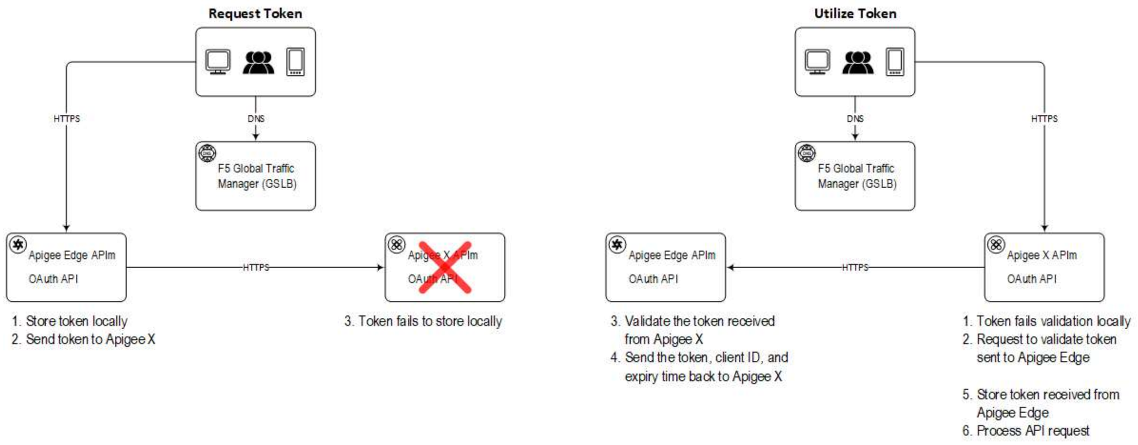
The metrics when calling the store token process as part post client flow are:

- Total time taken to generate the token: 21ms (20ms, 23ms, 17ms)
- Time taken in Apigee Edge for the service callout to store the token in Apigee X: < 1ms (As this is Async)
- Time taken in Apigee X for the Store Token service: 20ms

Back to the top

Token Generated in Apigee Edge and Not Stored in Apigee X Immediately

In this scenario, the token generated in Apigee Edge is only available in Apigee Edge to validate API calls and is not available in Apigee X because the request at the time of generation to store the token in Apigee X fails to process successfully.


Request Token
HTTPS
DNS
F5 Global Traffic Manager (GSLB)
Apigee Edge APIm
OAuth API
HTTPS
Apigee X APIm
OAuth API
1. Store token locally
2. Send token to Apigee X
3. Token fails to store locally
Utilize Token
DNS
HTTPS
F5 Global Traffic Manager (GSLB)
Apigee Edge APIm
OAuth API
HTTPS
Apigee X APIm
OAuth API
3. Validate the token received from Apigee X
4. Send the token, client ID, and expiry time back to Apigee X
1. Token fails validation locally
2. Request to validate token sent to Apigee Edge
5. Store token received from Apigee Edge
6. Process API request

When the token request is routed to the OAuth proxy in Apigee Edge, the proxy in Apigee Edge will generate the token, return the token to the client, and then attempt to send the token to Apigee X to store the token. However in this scenarion, Apigee X fails to store the token properly for whatever reason. When the subsequent traffic is received in Apigee X, Apigee X will attempt to validate the token. When that validation fails, Apigee X will then connect to Apigee Edge to validate the token. When that token is validated, Apigee X will then use the GenerateOAuth policy external authorization feature to store the token in Apigee X for the given client ID.

As part of the proof of concept, we implemented this approach and we have collected the following metrics for this transaction processing.

The metrics when calling the store token process as part of the generate flow are:

- Total time taken to generate the token: 50ms (30ms, 70ms, 64ms)
- Time taken in Apigee X to verify (it fails locally, calls Apigee Edge to validate, then stores the token in Apigee X): 272ms
- Time taken to validate the token from Apigee Edge for the first time using service callout: 187ms

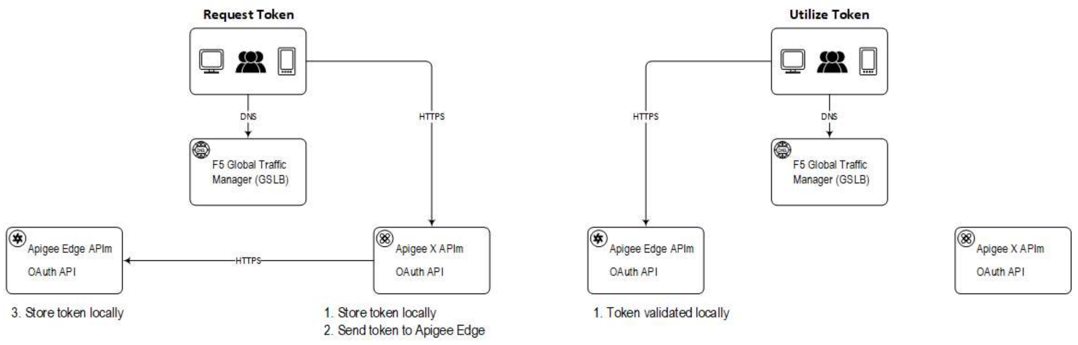
The metrics when calling the store token process as part post client flow are:

- Total time taken to generate the token: 22ms (20ms, 23ms, 17ms)
- Time taken in Apigee X to verify (it fails locally, calls Apigee Edge to validate, then stores the token in Apigee X): 187ms
- Time taken to validate the token from Apigee Edge for the first time using service callout: 166ms
  - Note : The flow in Apigee Edge completed in 4ms

Back to the top

Token Generated in Apigee X and Shared With Apigee Edge

In this scenario, the token generated in Apigee X is available in both Apigee X and Apigee Edge to validate API calls for both environments.



When the token request is routed to the OAuth proxy in Apigee X, the proxy in Apigee X will generate the token, return the token to the client, and then send the token to Apigee Edge to store the token. Apigee Edge will use the GenerateOAuth policy external authorization feature to store the token generated by Apigee X into Apigee Edge for the given client ID. When Apigee Edge receives a request, it now has the token stored to authorize the requested transactions.

As part of the proof of concept, we implemented this approach and we have collected the following metrics for this transaction processing.

The metrics when calling the store token process as part of the generate flow are:

- Total time taken to generate the token: 127ms (181ms, 92ms, 150ms)
- Time taken in Apigee X for the service callout to store the token in Apigee Edge: 60ms
- Time taken in Apigee Edge for the Store Token service: 21ms

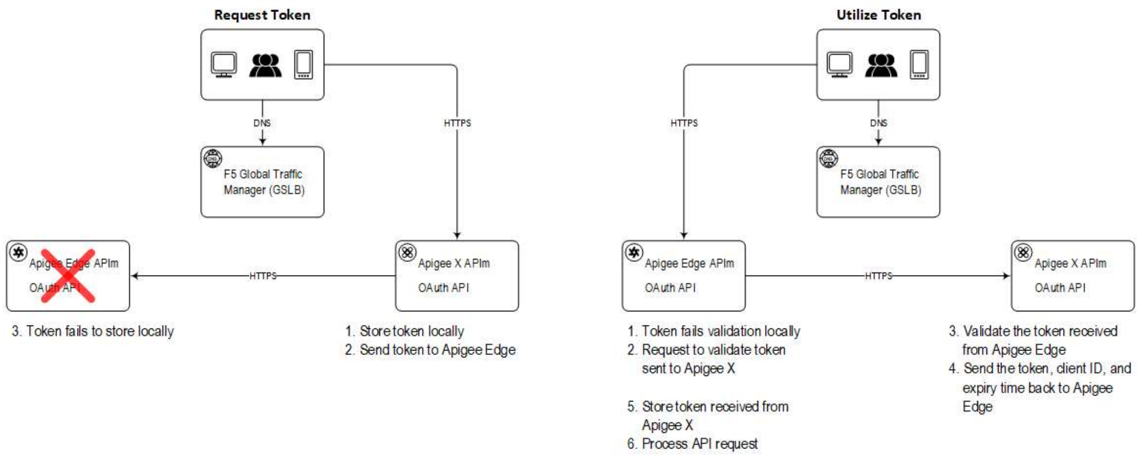
The metrics when calling the store token process as part post client flow are:

- Total time taken to generate the token: 80ms (120ms, 50ms, 27ms)
- Time taken in Apigee X for the service callout to store the token in Apigee Edge: 60ms
- Time taken in Apigee Edge for the Store Token service: 21ms

Back to the top

Token Generated in Apigee X and Not Stored in Apigee Edge Immediately

In this scenario, the token generated in Apigee X is only available in Apigee X to validate API calls and is not available in Apigee Edge because the request at the time of generation to store the token in Apigee Edge fails to process successfully.


Request Token
DNS
HTTPS
F5 Global Traffic Manager (GSLB)
Apigee Edge APIm
OAuth API
HTTPS
Apigee X APIm
OAuth API
3. Token fails to store locally
1. Store token locally
2. Send token to Apigee Edge
Utilize Token
HTTPS
DNS
F5 Global Traffic Manager (GSLB)
Apigee Edge APIm
OAuth API
HTTPS
Apigee X APIm
OAuth API
1. Token fails validation locally
2. Request to validate token sent to Apigee X
5. Store token received from Apigee X
6. Process API request
3. Validate the token received from Apigee Edge
4. Send the token, client ID, and expiry time back to Apigee Edge

When the token request is routed to the OAuth proxy in Apigee X, the proxy in Apigee X will generate the token, return the token to the client, and then attempt to send the token to Apigee Edge to store the token. However in this scenario, Apigee Edge fails to store the token properly for whatever reason. When the subsequent traffic is received in Apigee Edge, Apigee Edge will attempt to validate the token. When that validation fails, Apigee Edge will then connect to Apigee X to validate the token. When that token is validated, Apigee Edge will then use the GenerateOAuth policy external authorization feature to store the token in Apigee Edge for the given client ID.

As part of the proof of concept, we implemented this approach and we have collected the following metrics for this transaction processing.

The metrics when calling the store token process as part of the generate flow are:

- Total time taken to generate the token: 103ms (30ms, 70ms, 64ms)
- Time taken in Apigee Edge to verify (it fails locally, calls Apigee X to validate, then stores the token in Apigee Edge): 213ms
- Time taken to validate the token from Apigee X for the first time using service callout: 19ms

The metrics when calling the store token process as part post client flow are:

- Total time taken to generate the token: 16ms (25ms, 61ms, 16ms)
- Time taken in Apigee Edge to verify (it fails locally, calls Apigee X to validate, then stores the token in Apigee Edge): 19ms
- Time taken to validate the token from Apigee X for the first time using service callout: 142ms

Back to the top

OAuth APIs in Apigee Edge

API Name	Environment	Comments
FTS-APIM-OATUH2-TOKEN	test, qa, cat, cert, stage-prod, prod	Apigee Edge is the OAuth token provider
FTS-APIM-OAUTH2-V1	test, qa, cat, cert	PingFederate is the OAuth token provider

Back to the top

OAuth APIs in Apigee X

API Name	Environment	Comments
FTS-APIM-OATUH2-TOKEN	dev1, mtls-dev1, qa1	Apigee X is the OAuth token provider
FTS-APIM-OAUTH2-V1		PingFederate is the OAuth token provider

Back to the top

No labels